Oracle Tutorials

The architecture of Oracle RDBMS is client server architecture.

This architecture is supported by network communications based on the client communications with RDBMS (Database). Shared Memory created and allocated by Operating Syastem (OS) for inter process communication (IPC).

The network configuration is supported by

1. listener.ora
2. tnsnames.ora
3. sqlnet.ora

Based on the Oracle Docs, the process configuration:

Listener.ora

Listener for the RDBMS Confifuration is to be done to validate the incoming client process for communication with the database server.
The listener configuration depends on

01. protocol of the communication - TCP/IP or IPC
02. hostname - the server that is hosting the RDBMS
03. port - port on which the listener is hearing the taps by the client processes.

The host is identified by name or physical/virtual IP on TCP protocol.

This is achieved by the configuration of "Network Interface Controller (NIC)" configuration and activating the service

The incoming client process, initiates another in the server with the details of the listener protocol, host, and port (as identified in the configuration of oracle listener process) number. The server process carrying the process (TCP supported) port number establishes (with a shake hand with listener process). The listener after due validation communicates with the RDBMS Server and the server (RDBMS) creates a session within the database. That session excutes the DDL, DML commands as required by client and hands over the output to the client process and completes its role. Even after satisfying the query, the session can continue the session for further queries and communications.

tnsnames.ora

The client must to have Oracle Cleint installed and TNS configuration is completed withthe following details in tnsnames.ora

a. Address Of the RDBMS with

1. PROTOCOL - Oracle accepted protocol TCP
2. Host = <hostname>
3. Port = <port>

b. Connection information:

1. SERVICE_NAME = <service_name>

Service Name/s:

SERVICE_NAMES specifies one or more names by which clients can connect to the instance. The instance registers its service names with the listener.
If you do not qualify the names in this parameter with a domain, Oracle qualifies them with the value of the DB_DOMAIN parameter

Every Oracle instance registers its service names with the listener.

When a client requests a service, the listener determines which instances offer the requested service and routes the client to the appropriate instance.

In configuration specify multiple service names to distinguish among different uses of the same database.

The tnsnames.ora file is a configuration file that contains network service names mapped to connect descriptors for the local naming method, or net service names mapped to
listener protocol addresses.

A net service name is an alias mapped to a database network address contained in a connect descriptor. A connect descriptor contains the location of the listener through a protocol address and the service name of the database to which to connect. Clients and database servers (that are clients of other database servers) use the net service name when making a connection with an application.

By default, the tnsnames.ora file is located in the ORACLE_HOME/network/admin directory. Oracle Net will check the other directories for the configuration file. For example, the order checking the tnsnames.ora file is as follows:

The directory specified by the $TNS_ADMIN environment variable. If the file is not found in the directory specified, then it is assumed that the file does not exist.

If the $TNS_ADMIN environment variable is not set, then Oracle Net checks the $ORACLE_HOME/network/admin directory.

The client is instructed to connect to the protocol address of the first Oracle Connection Manager, as indicated by:

```
        (ADDRESS=(PROTOCOL=tcp)(HOST=host1)(PORT=1630))
```

The first Oracle Connection Manager is instructed to connect to the first protocol address of another Oracle Connection Manager. If the first protocol address fails, then it tries the second protocol address. This sequence is specified with the following configuration:

```
        (ADDRESS_LIST=
                (FAILOVER=on)
                (LOAD_BALANCE=off)
                (ADDRESS=(PROTOCOL=tcp)(HOST=<1st_hostname>)(PORT=<port_number_nnnn>))
                (ADDRESS=(PROTOCOL=tcp)(HOST=<2nd_hostname>)(PORT=<port_number_nnnn>))
        )
```

Oracle Connection Manager connects to the database service using the following protocol address:

```
        (ADDRESS=(PROTOCOL=tcp)(HOST=<3rd_hostname>)(PORT=<port_number_nnnn>))
```

The client load balancing among two Oracle Connection Managers and two protocol addresses.

The client is instructed to pick an ADDRESS_LIST at random and to fail over to the other if the chosen ADDRESS_LIST fails. This is indicated by the LOAD_BALANCE and FAILOVER parameters being set to on.

When an ADDRESS_LIST is chosen, the client first connects to Oracle Connection Manager, using the Oracle Connection Manager protocol address that uses port 1630 indicated for the ADDRESS_LIST.

Oracle Connection Manager then connects to the database service, using the protocol address indicated for the ADDRESS_LIST.

```
<tnsalias>=
(DESCRIPTION=
        (LOAD_BALANCE=on) -----------------------------------------                         # 1
        (FAILOVER=on)
(ADDRESS_LIST=
        (SOURCE_ROUTE=yes)
        (ADDRESS=(PROTOCOL=tcp)(HOST=<hostname_1>)(PORT=<port_nnnn>))  # 2
        (ADDRESS=(PROTOCOL=tcp)(HOST=<hostname_2>)(PORT=<port_nnnn>)))
(ADDRESS_LIST=
        (SOURCE_ROUTE=yes)
        (ADDRESS=(PROTOCOL=tcp)(HOST=<hostname_3>)(port=<port_nnnn>))
        (ADDRESS=(PROTOCOL=tcp)(HOST=<hostname_4>)(port=<port_nnnn>)))
(CONNECT_DATA=(SERVICE_NAME=<fqdn>)))                                  # 3
```

The client is instructed to pick an ADDRESS_LIST at random and to fail over to the other if the chosen ADDRESS_LIST fails. This is indicated by the LOAD_BALANCE and FAILOVER parameters being set to on.

When an ADDRESS_LIST is chosen, the client first connects to Oracle Connection Manager, using the Oracle Connection Manager protocol address that uses port 1630 indicated for the ADDRESS_LIST.

Oracle Connection Manager then connects to the database service, using the protocol address indicated for the ADDRESS_LIST.

SQLNet.ora configuration at the client and RDBMS Server:

Sqlnet.ora is a text file that provides SQL*Net with basic configuration details like tracing options, default domain, encryption, etc. This file can be found in the $ORACLE_HOME\network\admin directory.

The following parameters define sqlnet.ora profile:

ACCEPT_MD5_CERTS

ACCEPT_SHA1_CERTS

ADD_SSLV3_TO_DEFAULT

EXADIRECT_FLOW_CONTROL

EXADIRECT_RECVPOLL

DEFAULT_SDU_SIZE

DISABLE_OOB

DISABLE_OOB is a networking parameter of the sqlnet.ora file and is used to enable or disable Oracle Net to send or receive out-of-band break messages using urgent data provided by the underlying protocol.

DISABLE_OOB_AUTO

The DISABLE_OOB_AUTO networking parameter of the sqlnet.ora file checks the server path for out-of-band break messages support at the connection time.

HTTPS_SSL_VERSION

IPC.KEYPATH

NAMES.DEFAULT_DOMAIN

NAMES.DIRECTORY_PATH

NAMES.LDAP_AUTHENTICATE_BIND

NAMES.LDAP_CONN_TIMEOUT

NAMES.LDAP_PERSISTENT_SESSION

NAMES.NIS.META_MAP

RECV_BUF_SIZE

SDP.PF_INET_SDP

SEC_USER_AUDIT_ACTION_BANNER

SEC_USER_UNAUTHORIZED_ACCESS_BANNER

SEND_BUF_SIZE

SQLNET.ALLOW_WEAK_CRYPTO

Use the sqlnet.ora compatibility parameter SQLNET.ALLOW_WEAK_CRYPTO to configure your client-side network connection by reviewing the specified encryption and crypto-checksum algorithms.

SQLNET.ALLOW_WEAK_CRYPTO_CLIENTSUse the sqlnet.ora compatibility parameter SQLNET.ALLOW_WEAK_CRYPTO_CLIENTSto configure your server-side network connection by reviewing the specified encryption and crypto-checksum algorithms.

SQLNET.ALLOWED_LOGON_VERSION_CLIENT

SQLNET.ALLOWED_LOGON_VERSION_SERVER

SQLNET.AUTHENTICATION_SERVICES

SQLNET.CLIENT_REGISTRATION

SQLNET.CLOUD_USER

SQLNET.COMPRESSION

SQLNET.COMPRESSION_ACCELERATION

SQLNET.COMPRESSION_LEVELS

SQLNET.COMPRESSION_THRESHOLD

SQLNET.CRYPTO_CHECKSUM_CLIENT

SQLNET.CRYPTO_CHECKSUM_SERVER

SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT

SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER

SQLNET.DBFW_PUBLIC_KEY

SQLNET.DOWN_HOSTS_TIMEOUT

SQLNET.ENCRYPTION_CLIENT

The SQLNET.ENCRYPTION_CLIENT networking parameter turns encryption on for the client.

SQLNET.ENCRYPTION_SERVER

The SQLNET.ENCRYPTION_SERVER networking parameter turns encryption on for the database server.

SQLNET.ENCRYPTION_TYPES_CLIENT

Use the sqlnet.ora parameter SQLNET.ENCRYPTION_TYPES_CLIENT to list encryption algorithms for clients to use.

SQLNET.ENCRYPTION_TYPES_SERVER

Use the sqlnet.ora parameter SQLNET.ENCRYPTION_TYPES_SERVER to list the encryption algorithms for the database to use.

SQLNET.EXPIRE_TIME

SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS

The SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS parameter is used on the server-side to ignore the value set in SQLNET.ENCRYPTION_SERVER for TCPS connections (effectively disabling ANO encryption on the TCPS listener).

SQLNET.INBOUND_CONNECT_TIMEOUT

SQLNET.FALLBACK_AUTHENTICATION

SQLNET.KERBEROS5_CC_NAME

Use the sqlnet.ora parameter SQLNET.KERBEROS5_CC_NAME to specify the complete path name to the Kerberos credentials cache file.

SQLNET.KERBEROS5_CLOCKSKEW
SQLNET.KERBEROS5_CONF
SQLNET.KERBEROS5_CONF_LOCATION
SQLNET.KERBEROS5_KEYTAB
SQLNET.KERBEROS5_REALMS
SQLNET.KERBEROS5_REPLAY_CACHE
SQLNET.OUTBOUND_CONNECT_TIMEOUT
SQLNET.RADIUS_ALTERNATE
SQLNET.RADIUS_ALTERNATE_PORT
SQLNET.RADIUS_ALTERNATE_RETRIES
SQLNET.RADIUS_AUTHENTICATION
SQLNET.RADIUS_AUTHENTICATION_INTERFACE
SQLNET.RADIUS_AUTHENTICATION_PORT
SQLNET.RADIUS_AUTHENTICATION_RETRIES
SQLNET.RADIUS_AUTHENTICATION_TIMEOUT
SQLNET.RADIUS_CHALLENGE_RESPONSE
SQLNET.RADIUS_SECRET
SQLNET.RADIUS_SEND_ACCOUNTING
SQLNET.RECV_TIMEOUT

Use the sqlnet.ora parameter SQLNET.RECV_TIMEOUT to specify the duration of time that a database client or server should wait for data from a peer after establishing a connection.

SQLNET.SEND_TIMEOUT
SQLNET.URI

SQLNET.URI networking parameter of the sqlnet.ora file specifies a database client URI mapping on the web server.

SQLNET.USE_HTTPS_PROXY
SQLNET.WALLET_OVERRIDE
SSL_CERT_REVOCATION
SSL_CRL_FILE
SSL_CRL_PATH
SSL_CIPHER_SUITES
SSL_EXTENDED_KEY_USAGE
SSL_SERVER_DN_MATCH
SSL_VERSION

TCP.CONNECT_TIMEOUT
TCP.INVITED_NODES
TCP.NODELAY
TCP.QUEUESIZE
TCP.VALIDNODE_CHECKING

TNSPING.TRACE_DIRECTORY
TNSPING.TRACE_LEVEL

USE_CMAN
USE_DEDICATED_SERVER

WALLET_LOCATION
BEQUEATH_DETACH
It is a sqlnet.ora networking parameter handling POSIX signals for Linux and UNIX systems.

Session Validation:

Oracle validates any in coming session
Externally based on connect string.
The connect string shall have three (3) components:
a. username
b. password
c. @ -- reserved for differentiation of username, password from the database identifier.
d. database identifier (Service Name) or
system identifier (SID)
The Username should be pre-existing within the RDBMS.
The password (encrypted within the database) is validated for the existing username.

The database identier, service name is to be validated. The initialization db_name defines the database identifier.
If the db_name is not defined, instance_name becomes the database name by default.

If the initialization parameter service_names are defined then the the connect string is completed. The session is requesting connect string is validated confirming the username, password and database/service name.

Validation by Role of the trying to connect database:

Oracle have many roles for the users to be allowed to interact with the RDBMS. The roles should be of

1. system privileges - from creating session to creating and managing schema with creating the objects (including PL/SQL)
2. object privileges - creating and altering the objects and and performing DML activities.

Unless the user is NOT granted the "create session" system privilege, the user cannot create a session. His connection is validated but he cannot have a session within the RDBMS.

After creating a session (with the grant of create session system privilege by one of the database administration accounts such as SYS or SYSTEM) when an object like a table is to created, then the user has to have DDL privileges to create or alter objects.

Objects Privileges (DML privileges) on some other schema objects should be granted eithe by the owner or by the database administrator accounts.
If the user is not having any object privileges but is granted DDL privileges:

1. Unless the user is granted suitable object privileges and the object is not qualified with the schema (set of objects) then the query searches for results in the schema of the user.
2. If the session is looking for metadata of the objects granted the user is able to query the objects prefixed with "all_" and able to execute the PL/SQL objects prefixed with "dbms_".

That's how Oracle works. Wish you the best.